Commun. Comput. Phys.
doi: 10.4208/cicp.210612.240113a

Vol. **14**, No. 4, pp. 1073-1093
October 2013

A Fast Accurate Boundary Integral Method for Potentials on Closely Packed Cells

Wenjun Ying[1] and J. Thomas Beale[2,*]

[1] *Shanghai Jiao Tong University, Department of Mathematics and Institute of Natural Sciences, Minhang, Shanghai 200240, P.R. China.*
[2] *Duke University, Department of Mathematics, Durham, NC 27708-0320, USA.*

Received 21 June 2012; Accepted (in revised version) 24 January 2013

Available online 10 May 2013

Abstract. Boundary integral methods are naturally suited for the computation of harmonic functions on a region having inclusions or cells with different material properties. However, accuracy deteriorates when the cell boundaries are close to each other. We present a boundary integral method in two dimensions which is specially designed to maintain second order accuracy even if boundaries are arbitrarily close. The method uses a regularization of the integral kernel which admits analytically determined corrections to maintain accuracy. For boundaries with many components we use the fast multipole method for efficient summation. We compute electric potentials on a domain with cells whose conductivity differs from that of the surrounding medium. We first solve an integral equation for a source term on the cell interfaces and then find values of the potential near the interfaces via integrals. Finally we use a Poisson solver to extend the potential to a regular grid covering the entire region. A number of examples are presented. We demonstrate that increased refinement is not needed to maintain accuracy as interfaces become very close.

AMS subject classifications: 35J05, 65N06, 65N38

Key words: Laplace equation, boundary integral method, fast multipole method.

1 Introduction

A wide range of biological problems lead to models involving a potential function in tissue with a number of closely packed cells. Recent applications include gene transfection [14, 15], electrochemotherapy of tumors [20] and cardiac defibrillation [1]. Our interest in the problem is mainly motivated by studies of the electrical response of biological cells under field stimulation [25], which can be described by harmonic potential

*Corresponding author. *Email addresses:* wying@sjtu.edu.cn (W.-J. Ying), beale@math.duke.edu (J. T. Beale)

functions on a domain consisting of many cells and an extracellular region with different conductivities. These potentials are naturally expressed as integrals on the cell boundaries. Boundary integral methods are well suited for the computation of such problems, but they require special care when the cell boundaries are close. We present a method which is designed to handle such cases accurately and efficiently. We focus here on the model of electrical stimulation of cells in two dimensions, but similar computational issues occur in other applications. For example, the motion of drops of one viscous fluid in another, or the fluid motion of vesicles, such as blood cells, is often modeled by Stokes flow, leading to a related integral formulation, again with many components embedded in a surrounding medium [16, 22, 23, 26, 27].

The electric potential problem is formulated in Sections 2 and 3. Since the potential is a harmonic function inside each cell and in the exterior region, with prescribed boundary conditions at the cell interfaces, it is natural to write the potential as a sum of single and double layer potentials on the cell boundaries Γ_k, $k=1,\cdots,K$, and evaluate the integrals directly. In principle this is routine if the point of evaluation $\mathbf{x}$ is away from Γ_k. It is also not difficult in this two-dimensional setting if $\mathbf{x}\in\Gamma_k$. However, if, for example, Γ_1 and Γ_2 are close and $\mathbf{x}\in\Gamma_1$, then the integral on Γ_2 is nearly singular, so that a standard quadrature rule becomes inaccurate when the distance is small. It is therefore desirable to use a method of quadrature which is accurate, uniformly with respect to the point of evaluation, without requiring a large amount of extra work. A method with these features was developed in [5] and is used here. Briefly, the singularity in the integral kernel is regularized on a scale comparable to the grid size, and a standard quadrature is used for the regularized integral. Analytical corrections are then added for the errors due to regularization and discretization. It is not necessary to use special quadrature points depending on the point of evaluation; the method is almost as efficient as for a smooth integrand. The present work makes practical use of this integration method in the case where several interfaces are close to each other. The quadrature and correction formulas are given in detail in Section 4. A similar method for layer potentials on surfaces was developed in [4].

To solve for the potential, we first solve an integral equation for a source term on the cell boundaries. We then compute the potential at grid points covering the region of interest. To compute the integrals directly would require a large computational cost, especially if the number of cells is large. For this reason we use a version of the fast multipole method for the interaction of points which are well separated; the effect of the regularization can be neglected in this case. The nearby interactions are summed directly, using the regularized kernel. This procedure is explained in Section 5. The regularization must be used within a large enough radius for accuracy, but it imposes a lower bound on the capacity in the tree structure for the fast summation. Guidelines for the choice of parameters are given to balance the accuracy and efficiency. A similar procedure was used in the context of regularized point vortices in [10].

After solving the integral equation, we evaluate the potential at grid points near the cell boundaries. These values are again given by nearly singular integrals, which are

computed in a similar manner using the multipole method. From these we can obtain the values at all grid points with a fast Poisson solver, using a method introduced in [13] and described in Section 6. The operation count for the full method is at worst roughly proportional to the total number of grid points in the region and on the cell boundaries.

The difficulty in computing nearly singular integrals has long been recognized, e.g., [2, Sec. 7.2.1]. For viscous fluid flow with interfaces, modeled by the Stokes equations, the interfacial motion can be determined by computing the velocity only on the interfaces. The velocity is written in terms of boundary integrals using the fundamental solution of Stokes flow. For flow with many drops or vesicles, the integral on one interface will be nearly singular when evaluated at a point on a nearby interface [22, 23, 26, 27]. A method similar to that developed here could be used in such cases. Boundary integrals are often computed using boundary element methods. In this approach integrals are computed on triangular elements, usually using coordinate transformations. A recent, comprehensive mathematical treatment of boundary element methods is given in [19]. The method [24] for computing layer potentials on surfaces, using coordinate charts rather than boundary elements, included a special treatment of the nearly singular case.

The problem is formulated in Section 2. In Section 3 the integral expression for the solution is derived in terms of a source term q on the boundary, and an integral equation for q is stated. In Section 4 formulas are given for the evaluation of the nearly singular integrals. The combination of the fast multipole method with the regularization is described in Section 5. The full method is outlined in Section 6, along with a description of the method of [13]. Numerical examples for the interface problem on one, three and twenty closely packed cells are presented in Section 7 and briefly discussed in Section 8. The integral equation is derived in the Appendix.

2 Interface problem

Let $\Omega_i \subset \mathbb{R}^2$ be a bounded open set with smooth boundary, which may have multiple disconnected components. Let $\Omega_e = \mathbb{R}^2 \setminus \bar{\Omega}_i$ be the unbounded, complementary domain. Let Γ be the interface, the common boundary of Ω_i and Ω_e. When the interface Γ has multiple components, we write $\Gamma = \bigcup_{k=1}^{K} \Gamma_k$, and assume each component Γ_k is a simple closed curve.

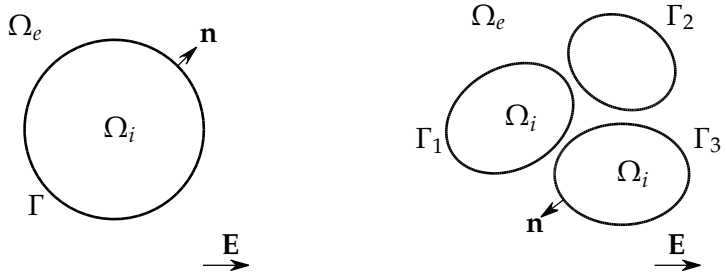
Let $\mathbf{x} = (x_1, x_2)^{\mathrm{T}} \in \mathbb{R}^2$ be a point in space. Let $\Phi_i(\mathbf{x})$ and $\Phi_e(\mathbf{x})$ be two unknown potential functions on Ω_i and Ω_e, respectively. Assume the conductivities σ_i and σ_e on Ω_i and Ω_e are constant but distinct ($\sigma_i \neq \sigma_e$). The potential functions satisfy the Laplace equation

$$\triangle \Phi_i(\mathbf{x}) = 0 \qquad \text{in } \Omega_i, \tag{2.1a}$$

$$\triangle \Phi_e(\mathbf{x}) = 0 \qquad \text{in } \Omega_e. \tag{2.1b}$$

Let

$$\Phi(\mathbf{x}) = \begin{cases} \Phi_i(\mathbf{x}), & \mathbf{x} \in \Omega_i, \\ \Phi_e(\mathbf{x}), & \mathbf{x} \in \Omega_e. \end{cases}$$



(a) (b)

Figure 1: Interface Γ separating subdomains Ω_i and Ω_e: (a) a circular interface; (b) a three-component interface.

In general, the function $\Phi(\mathbf{x})$ is discontinuous across the interface Γ. Let

$$\Phi_i(\mathbf{x})-\Phi_e(\mathbf{x})=V_m(\mathbf{x}) \qquad \text{on } \Gamma, \tag{2.2}$$

where $V_m(\mathbf{x})$ will be known. Assume that

$$\sigma_i\frac{\partial\Phi_i(\mathbf{x})}{\partial\mathbf{n}_\mathbf{x}}-\sigma_e\frac{\partial\Phi_e(\mathbf{x})}{\partial\mathbf{n}_\mathbf{x}}=J_m \qquad \text{on } \Gamma, \tag{2.3}$$

with $\mathbf{n}_\mathbf{x}$ being the unit normal pointing from the bounded domain Ω_i to the unbounded domain Ω_e. For a realistic problem such as the heart tissue, J_m is typically equal to zero [12,25]. This term is added mainly for testing. Let $\mathbf{E}=(E_1,E_2)^\mathrm{T}\in\mathbb{R}^2$ be a given vector. We assume the potential function $\Phi_e(\mathbf{x})$ satisfies the far field condition

$$\Phi_e(\mathbf{x})\to-\mathbf{E}\cdot\mathbf{x} \qquad \text{as } |\mathbf{x}|\to\infty. \tag{2.4}$$

The interface problem (2.1a)-(2.4) may describe the electrical response of biological cells under an external field stimulation with $\mathbf{E}$ being the applied electric field [12,25].

3 Boundary integral equation

We will express Φ in terms of double and single layer potentials of the form

$$u(\mathbf{x})=\int_\Gamma\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial\mathbf{n}_\mathbf{y}}f(\mathbf{y})ds_\mathbf{y}, \qquad v(\mathbf{x})=\int_\Gamma G(\mathbf{x}-\mathbf{y})q(\mathbf{y})ds_\mathbf{y} \tag{3.1}$$

with some density functions f and q. Here $G(\mathbf{x})=(2\pi)^{-1}\log|\mathbf{x}|$ is the fundamental solution of the Laplace equation in $\mathbb{R}^2$ and $s_\mathbf{y}$ is the arc length parameter of the interface Γ.

We recall that u has a jump discontinuity at Γ,

$$\begin{cases} u_i(\mathbf{x}) = \dfrac{1}{2}f(\mathbf{x}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_\mathbf{y}} f(\mathbf{y})\, ds_\mathbf{y}, \\ u_e(\mathbf{x}) = -\dfrac{1}{2}f(\mathbf{x}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_\mathbf{y}} f(\mathbf{y})\, ds_\mathbf{y}, \end{cases} \tag{3.2}$$

while $\partial u/\partial \mathbf{n}$ is continuous across Γ. Also v is continuous at Γ but $\partial v/\partial \mathbf{n}$ has a jump,

$$\begin{cases} \dfrac{\partial v_i(\mathbf{x})}{\partial \mathbf{n}_\mathbf{x}} = -\dfrac{1}{2}q(\mathbf{x}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{n}_\mathbf{x}} q(\mathbf{y})\, ds_\mathbf{y}, \\ \dfrac{\partial v_e(\mathbf{x})}{\partial \mathbf{n}_\mathbf{x}} = \dfrac{1}{2}q(\mathbf{x}) + \displaystyle\int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{n}_\mathbf{x}} q(\mathbf{y})\, ds_\mathbf{y}. \end{cases} \tag{3.3}$$

Now, assuming the solution Φ of the interface problem above exists, let

$$q(\mathbf{x}) = \frac{\partial \Phi_i(\mathbf{x})}{\partial \mathbf{n}_\mathbf{x}} - \frac{\partial \Phi_e(\mathbf{x})}{\partial \mathbf{n}_\mathbf{x}} \qquad \text{on } \Gamma. \tag{3.4}$$

Then the potential function $\Phi(\mathbf{x})$ can be represented as

$$\Phi(\mathbf{x}) = \int_\Gamma \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_\mathbf{y}} V_m(\mathbf{y})\, ds_\mathbf{y} - \int_\Gamma G(\mathbf{x}-\mathbf{y}) q(\mathbf{y})\, ds_\mathbf{y} - \mathbf{E}\cdot\mathbf{x}. \tag{3.5}$$

According to the properties above, this expression for Φ will have the jumps prescribed in (2.2) and (3.4). The unknown density $q(\mathbf{x})$ in (3.5) is determined by the interface condition (2.3).

Let $\mathbf{t}_\mathbf{x} = (x_1'(s), x_2'(s))^T$ be the unit tangent along the interface, so that the unit outward normal $\mathbf{n}_\mathbf{x} = (x_2'(s), -x_1'(s))^T$. From the continuity properties of the single and double layer potentials and the interface condition (2.3), we get the boundary integral equation

$$\frac{1}{2}q(\mathbf{x}) + \mu \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{n}_\mathbf{x}} q(\mathbf{y})\, ds_\mathbf{y} = \mu \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{t}_\mathbf{x}} \frac{\partial V_m(\mathbf{y})}{\partial \mathbf{t}_\mathbf{y}}\, ds_\mathbf{y} - \mu \mathbf{E}\cdot\mathbf{n}_x + \jmath_m \tag{3.6}$$

with $\mu = (\sigma_e - \sigma_i)/(\sigma_e + \sigma_i) \in (-1,1)$ and $\jmath_m = J_m/(\sigma_i + \sigma_e)$. Here the second integral is meant in the principal value sense; the leading singularity is removed in the computational method of Section 4. A brief derivation of this equation is given in the Appendix. The integral equation (3.6) can be re-written concisely as

$$\frac{1}{2}q + \mu \mathcal{K} q = \mu g + \jmath_m \qquad \text{on } \Gamma, \tag{3.7}$$

with

$$g(\mathbf{x}) = \mathcal{L} V_m - \mathbf{E}\cdot\mathbf{n}_\mathbf{x} \qquad \text{on } \Gamma,$$

where $\mathcal{K}$ and $\mathcal{L}$ are the integral operators defined on the interface

$$(\mathcal{K}q)(\mathbf{x}) = \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{n}_\mathbf{x}} q(\mathbf{y})\, ds_\mathbf{y} \qquad \text{on } \Gamma,$$
$$(\mathcal{L}V_m)(\mathbf{x}) = \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{t}_\mathbf{x}} \frac{\partial V_m(\mathbf{y})}{\partial \mathbf{t}_\mathbf{y}} ds_\mathbf{y} \qquad \text{on } \Gamma.$$

Integral equations such as (3.7) are often solved by the biconjugate gradient method [21, 27] or the generalized minimal residual (GMRES) method [17, 18, 22]. In this work, we solve the integral equation (3.7) by simple iteration. The spectrum of the operator $\mathcal{K}$ is contained in the interval $-\frac{1}{2} < \lambda \le \frac{1}{2}$, and consequently the simple iteration

$$q_{n+1} = (1-\beta) q_n + 2\beta(\mu g + \jmath_m - \mu \mathcal{K} q_n) \qquad \text{for} \quad n = 0, 1, 2, \cdots \tag{3.8}$$

converges to the exact solution for $0 < \beta < 2/(1+\mu)$.

4 Evaluation of boundary integrals

To compute the single layer potential in (3.5) we will integrate by parts so that the kernel is the tangential derivative of G. We first note that for each interface component Γ_k

$$\int_{\Gamma_k} q(\mathbf{y})\, ds_\mathbf{y} = 0,$$

where q is defined in (3.4). To see this, we first note that the integral of $\partial \Phi_i(\mathbf{y})/\partial \mathbf{n}_\mathbf{y}$ is zero, using the divergence theorem in the interior of Γ_k. But the same is true for $\partial \Phi_e(\mathbf{y})/\partial \mathbf{n}_\mathbf{y}$ because of the flux condition (2.3), and therefore also for q. Thus

$$Q(r) = \int_0^r q(\mathbf{y})\, ds_\mathbf{y}$$

is a periodic function of the arc length, and we can write

$$\int_{\Gamma_k} G(\mathbf{y}-\mathbf{x}) q(\mathbf{y})\, ds_\mathbf{y} = -\int_{\Gamma_k} \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{t}_\mathbf{y}} Q(s_\mathbf{y})\, ds_\mathbf{y}.$$

Now to evaluate $\Phi(\mathbf{x})$ in (3.5) we will compute boundary integrals of the form

$$u^{(k)}(\mathbf{x}) = \int_{\Gamma_k} \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_\mathbf{y}} f(\mathbf{y})\, ds_\mathbf{y}, \tag{4.1a}$$
$$v^{(k)}(\mathbf{x}) = \int_{\Gamma_k} \frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{t}_\mathbf{y}} f(\mathbf{y})\, ds_\mathbf{y} \tag{4.1b}$$

on each interface component Γ_k. When $\mathbf{x}$ is far away from Γ_k we replace the integral by a trapezoidal sum and use the fast summation method described in the next section. For

$\mathbf{x}\in\Gamma_k$, the kernel in (4.1a) is smooth, but (4.1b) is a principal value integral. When $\mathbf{x}$ is on or close to Γ_k, we compute the integrals using the method developed in [5]. We summarize the procedure here, with slight differences in notation. For both (4.1a) and (4.1b) the singularity or near singularity is reduced by subtraction; in particular, the integrand in (4.1b) becomes integrable for $\mathbf{x}\in\Gamma_k$. The singular integral is regularized with a length parameter δ, usually chosen to be of the order of the grid size. The integral is replaced by a sum, and two corrections are added. The resulting approximation to the integral is accurate to about $\mathcal{O}(\delta^3)$. Next we present the specific formulas for computing $u^{(k)}$ and $v^{(k)}$.

Assume each Γ_k is parametrized as $\mathbf{y}=\mathbf{y}(\alpha)$, $0\le\alpha\le 2\pi$, and the α-segment is partitioned into M_k segments with grid points $\alpha_j^{(k)}=2\pi j/M_k$, $1\le j\le M_k$. We express $u^{(k)}$ as

$$\begin{aligned} u^{(k)}(\mathbf{x}) &= \int_{\Gamma_k}\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_{\mathbf{y}}} f(\mathbf{y})\,ds_{\mathbf{y}} \\ &= \int_0^{2\pi} N(\alpha)\cdot\nabla G(\mathbf{y}(\alpha)-\mathbf{x})[f(\alpha)-f(\alpha_0^{(k)})]\,d\alpha+\chi(\mathbf{x})f(\alpha_0^{(k)}). \end{aligned} \tag{4.2}$$

Here $N(\alpha)=(y_2'(\alpha),-y_1'(\alpha))$; $\chi(\mathbf{x})=1$ for $\mathbf{x}$ inside Γ_k, $\chi(\mathbf{x})=0$ for $\mathbf{x}$ outside; and $\mathbf{y}(\alpha_0^{(k)})$ is the closest point on Γ_k to $\mathbf{x}$, so that $\mathbf{x}=\mathbf{y}_0+b\mathbf{n}_0$, for some b, where $\mathbf{y}_0=\mathbf{y}(\alpha_0^{(k)})$ and $\mathbf{n}_0=\mathbf{n}_{\mathbf{y}_0}$. Replacing ∇G with a regularized version

$$\begin{aligned} \nabla G_\delta(\mathbf{y}(\alpha)-\mathbf{x}) &= (1-e^{-r^2/\delta^2})\nabla G(\mathbf{y}(\alpha)-\mathbf{x}) \\ &= \frac{1}{2\pi r^2}(1-e^{-r^2/\delta^2})(\mathbf{y}(\alpha)-\mathbf{x}), \end{aligned} \tag{4.3}$$

with $r=|\mathbf{y}(\alpha)-\mathbf{x}|$, we approximate $u_k(\mathbf{x})$ by

$$u_h^{(k)}(\mathbf{x})=\frac{2\pi}{M_k}S_{u,1}^{(k)}(\mathbf{x})-\frac{2\pi}{M_k}f(\alpha_0^{(k)})S_{u,0}^{(k)}(\mathbf{x})+\chi(\mathbf{x})f(\alpha_0^{(k)})+T_{u,1}+T_{u,2}$$

with

$$S_{u,1}^{(k)}(\mathbf{x})=\sum_{j=1}^{M_k}N(\alpha_j^{(k)})\cdot\nabla G_\delta(\mathbf{y}(\alpha_j^{(k)})-\mathbf{x})f(\alpha_j^{(k)}), \tag{4.4a}$$

$$S_{u,0}^{(k)}(\mathbf{x})=\sum_{j=1}^{M_k}N(\alpha_j^{(k)})\cdot\nabla G_\delta(\mathbf{y}(\alpha_j^{(k)})-\mathbf{x}). \tag{4.4b}$$

The terms $T_{u,1}$ and $T_{u,2}$ are corrections for regularization and discretization, derived in [5]. The first is

$$T_{u,1}=-\delta^2(4\pi)^{-1}\eta\left(\sqrt{\pi}e^{-\eta^2}-\pi|\eta|\mathrm{erfc}|\eta|\right)\left(|\mathbf{y}_0'|^{-2}f_0''-|\mathbf{y}_0'|^{-4}(\mathbf{y}_0''\cdot\mathbf{y}_0')f_0'\right),$$

where $\eta = b/\delta$, $\mathbf{y}_0' = \mathbf{y}'(\alpha_0^{(k)})$, and similarly for $\mathbf{y}_0''$, f_0', f_0''. The second is

$$T_{u,2} = -\frac{hf_0'\eta\sigma}{2}\sum_{n=1}^{\infty}\sin(2n\pi\alpha_0^{(k)}/h)E^+(\eta, n\pi\sigma)$$

with $\sigma = \delta/(h|\mathbf{x}_0'|)$ and

$$E^{\pm}(\eta,\zeta) = e^{2\eta\zeta}\mathrm{erfc}(\eta+\zeta) \pm e^{-2\eta\zeta}\mathrm{erfc}(-\eta+\zeta).$$

The sum in $T_{u,2}$ is rapidly convergent.

The treatment of $v^{(k)}$ is similar:

$$\begin{aligned} v^{(k)}(\mathbf{x}) &= \int_{\Gamma_k}\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{t}_{\mathbf{y}}} f(\mathbf{y})\,ds_{\mathbf{y}} \\ &= \int_0^{2\pi}\mathbf{y}'(\alpha)\cdot\nabla G(\mathbf{y}(\alpha)-\mathbf{x})[f(\alpha)-f(\alpha_0^{(k)})]\,d\alpha. \end{aligned} \tag{4.5}$$

We approximate $v^{(k)}(\mathbf{x})$ by

$$v_h^{(k)}(\mathbf{x}) = \frac{2\pi}{M_k}S_{v,1}^{(k)}(\mathbf{x}) - \frac{2\pi}{M_k}f(\alpha_0^{(k)})S_{v,0}^{(k)}(\mathbf{x}) + T_{v,1} + T_{v,2}.$$

with

$$S_{v,1}^{(k)}(\mathbf{x}) = \sum_{j=1}^{M_k}\mathbf{y}'(\alpha_j^{(k)})\cdot\nabla G_\delta(\mathbf{y}(\alpha_j^{(k)})-\mathbf{x})f(\alpha_j^{(k)}), \tag{4.6a}$$

$$S_{v,0}^{(k)}(\mathbf{x}) = \sum_{j=1}^{M_k}\mathbf{y}'(\alpha_j^{(k)})\cdot\nabla G_\delta(\mathbf{y}(\alpha_j^{(k)})-\mathbf{x}). \tag{4.6b}$$

The corrections $T_{v,1}$ and $T_{v,2}$ are

$$T_{v,1} = -\delta(2\pi)^{-1}f_0'|\mathbf{y}_0'|^{-1}(1+\kappa_0\eta\delta/2)\left(\sqrt{\pi}e^{-\eta^2} - \pi|\eta|\mathrm{erfc}|\eta|\right), \tag{4.7a}$$

$$T_{v,2} = hf_0'\sum_{n=1}^{\infty}\cos(2n\pi\alpha_0^{(k)}/h)\left(-\frac{\eta\sigma}{2}E^-(\eta,n\pi\sigma) + \frac{\sigma}{\sqrt{\pi}}e^{-\eta^2}e^{-n^2\pi^2\sigma^2}\right), \tag{4.7b}$$

where κ_0 is the curvature at $\mathbf{y}_0$, defined by $\mathbf{y}_{ss} = \kappa_0\mathbf{n_0}$.

For the integral equation (3.6) we need to compute partial derivatives of the single layer potential. To do this we rewrite them in integrals in the form of $u^{(k)}$ and $v^{(k)}$ above,

$$\begin{aligned} \int_\Gamma\frac{\partial G}{\partial x_1}(\mathbf{y}-\mathbf{x})q(\mathbf{y})\,ds_{\mathbf{y}} &= -\int_\Gamma\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_{\mathbf{y}}}q(\mathbf{y})y_2'(s)\,ds_{\mathbf{y}} \\ &\quad -\int_\Gamma\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{t}_{\mathbf{y}}}q(\mathbf{y})y_1'(s)\,ds_{\mathbf{y}}, \\ \int_\Gamma\frac{\partial G}{\partial x_2}(\mathbf{y}-\mathbf{x})q(\mathbf{y})\,ds_{\mathbf{y}} &= \int_\Gamma\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{n}_{\mathbf{y}}}q(\mathbf{y})y_1'(s)\,ds_{\mathbf{y}} \\ &\quad -\int_\Gamma\frac{\partial G(\mathbf{y}-\mathbf{x})}{\partial \mathbf{t}_{\mathbf{y}}}q(\mathbf{y})y_2'(s)\,ds_{\mathbf{y}}. \end{aligned}$$

These formulas are easily justified for $\mathbf{x} \notin \Gamma$, and they also hold for $x \in \Gamma$, again in the principal value sense, by averaging the limiting values on the two sides. These integrals can be evaluated in the manner just described.

5 The fast multipole method

For each fixed target point $\mathbf{x}$, the summations (4.4a)-(4.4b) and (4.6a)-(4.6b) involve computational work on the order of $J = \sum_{k=1}^{K} M_k$. Thus the evaluation of boundary integrals at $\mathcal{O}(J)$ target points requires computational work of $\mathcal{O}(J^2)$. The fast multipole method, which was introduced by Rokhlin and Greengard [9] and has been acclaimed as one of the top-ten algorithms of the 20^{th} century [7], is applied to reduce to the complexity of these summations from $\mathcal{O}(J^2)$ to $\mathcal{O}(J \log J)$.

The sums (4.4a) and (4.6a) for the different interface components Γ_k can be combined. That is, the total summations

$$S_{u,1}(\mathbf{x}) = \sum_{k=1}^{K} S_{u,1}^{(k)}(\mathbf{x}) = \sum_{k=1}^{K} \sum_{j=1}^{M_k} N_k(\alpha_j^{(k)}) \cdot \nabla G_\delta(\mathbf{y}(\alpha_j^{(k)}) - \mathbf{x}) f(\alpha_j^{(k)}),$$

$$S_{v,1}(\mathbf{x}) = \sum_{k=1}^{K} S_{v,1}^{(k)}(\mathbf{x}) = \sum_{k=1}^{K} \sum_{j=1}^{M_k} \mathbf{y}'(\alpha_j^{(k)}) \cdot \nabla G_\delta(\mathbf{y}(\alpha_j^{(k)}) - \mathbf{x}) f(\alpha_j^{(k)}),$$

each of which is in the form

$$w(\mathbf{x}_i) = \sum_{j=1}^{J} A(\mathbf{y}_j, \mathbf{x}_i) q(\mathbf{y}_j), \tag{5.1}$$

are computed at once by the fast multipole method. However, for the sums (4.4b) and (4.6b), we have to apply the fast multipole method separately for each component Γ_k, since the multiplication of the sum by $f(\alpha_0^{(k)})$, which depends on the target point $\mathbf{x}$, makes the total sum over the different components fail to be in the form (5.1).

In two space dimensions, the double layer potential (4.1a) and the tangential derivative (4.1b) of the single layer potential happen to be the real and imaginary parts of a Cauchy integral if the spatial coordinates are treated as complex variables. The fast multipole method uses the multipole expansions for the kernel of the Cauchy integral to group sources that lie close together, but far away from the target point, and treat them as if they are a single source.

In the fast multipole method, each point is associated with a list of near field interaction points and a list of far field interaction points, the latter of which is not explicitly stored. The part of the summation due to contributions from the far field points is computed by the multipole expansions and translations, including far field expansion, multipole to multipole translation from coarse to fine level and multipole to local translation at the same level [9].

Since it is not clear if a simple multipole expansion can be used for the regularized kernel, in this work the standard kernel is used in the far field calculations with the fast multipole method while the regularized version of the fundamental solution is used only for those grid nodes $\mathbf{y}(\alpha_j)$ which are on the near neighbor interaction list of the target point $\mathbf{x}$. The error due to the approximation should be negligible when the smoothing parameter δ is sufficiently small since in this case the exponential function e^{-r^2/δ^2} with $r=|\mathbf{y}-\mathbf{x}|$ in (4.3) is almost zero for source points which are not in the near neighbor interaction list.

The fast multipole method implemented in this work represents the discrete points by a quad-tree data structure. Each quad-tree node represents a rectangular patch and is allowed to contain at most C source points; C is called the capacity parameter of the quad-tree structure.

Next we give a method to roughly estimate the capacity parameter C that is reasonably large enough for the multipole summation to be accurate as well as efficient. For a source point $\mathbf{y}$ which is not in the near neighbor interaction list of a target point $\mathbf{x}$, its distance $r=|\mathbf{y}-\mathbf{x}|$ away from the target point is on average about $Ch_k/2$ with $h_k=\mathcal{L}_k/J$, assuming the source points locally are aligned along nearly straight lines. Here $\mathcal{L}_k$ is the arc length of the k^{th} interface component Γ_k. In order for the error in replacing the standard kernel with the regularized one (4.3) to be within some tolerance parameter ϵ_{fmm}, the capacity parameter C needs to satisfy

$$\exp\Big\{-\Big(\frac{Ch_k}{2\delta}\Big)^2\Big\}<\epsilon_{\text{fmm}}$$

or

$$C>\frac{2\delta}{h_k}\sqrt{|\ln\epsilon_{\text{fmm}}|}.$$

For $\epsilon_{\text{fmm}}=10^{-8}$, we have $\sqrt{|\ln\epsilon_{\text{fmm}}|}\approx 4.29$ and an estimate for the lower bound of the capacity parameter, $C>8.58\times\delta/h_k$. This will guide us in the selection of the capacity parameter for the fast multipole summation. However, the practical lower bound of the capacity parameter varies due to the local symmetry and curvature of the interface.

6 Solution on Cartesian grid

After the unknown density is solved from the boundary integral equation with the method described in the previous sections, in principle the value of the potential to the interface problem at any point can be obtained by evaluating the boundary integrals. However, in this work we avoid computing the potential at an arbitrary point with the boundary integral evaluation method. Instead, a fast Fourier transform (FFT) based Poisson solver is used to obtain values of the potential at nodes of a Cartesian grid on a rectangle box, which embeds the interface.

We solve for values of the harmonic potential Φ, described in Section 2, at the Cartesian grid nodes, with prescribed V_m and $\mathbf{E}$, in three steps:

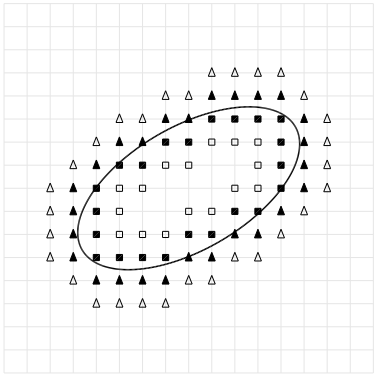


Figure 2: A Cartesian grid with the irregular grid nodes being marked

Step 1. We solve the integral equation (3.6) for the interfacial density q. We compute the integrals as in Section 4, with far field interactions computed by the fast multipole method as described in Section 5.

Step 2. We choose as the computational domain a rectangular box $\mathcal{B}$ enclosing the cells bounded by the Γ_k and introduce a Cartesian grid. We compute the potential Φ at all grid nodes in a neighborhood of the interface Γ, from the formula (3.5), again using the methods of Sections 4 and 5. The set of grid nodes is described below and illustrated in Fig. 2. We also compute Φ at the grid points on the boundary of $\mathcal{B}$; this is routine since these points are away from Γ.

Step 3. We form the discrete Laplacian for Φ near Γ, extend it to be zero elsewhere in $\mathcal{B}$, and then invert the discrete Laplacian to obtain values for Φ at all grid points in $\mathcal{B}$.

Step 3 uses a method introduced by A. Mayo [13] which we now describe. We use the usual discrete Laplacian

$$\Delta_h\Phi = \left(\Phi_{i-1,j}+\Phi_{i,j-1}+\Phi_{i+1,j}+\Phi_{i,j+1}-4\Phi_{i,j}\right)/h^2$$

with h be the mesh parameter of the Cartesian grid. The nodes adjacent to the interface Γ (i.e., those whose discrete Laplacian crosses the interface) are called irregular grid nodes, and the others are called regular grid nodes. The values of the boundary integrals at the irregular grid nodes (the solid marked points in Fig. 2), their neighboring regular nodes (the other marked points in Fig. 2) and the boundary nodes of the regular box are computed by evaluating the integral (3.5). From these values we can form the discrete Laplacian at the irregular nodes. Away from the interface, Φ is smooth and $\Delta\Phi = 0$; therefore, at the regular nodes, $\Delta_h\Phi = \mathcal{O}(h^2)$. We now form an approximation to $\Delta_h\Phi$ for all grid nodes in $\mathcal{B}$, using the computed values at the irregular nodes and zero at the regular nodes. Finally, having an expression for $\Delta_h\Phi$ throughout $\mathcal{B}$ and computed values of Φ on the boundary, we solve the discrete Poisson equation,

$$\Delta_h\Phi_{i,j} = \begin{cases} 0, & \text{at regular grid points,} \\ \Delta_h w_{i,j}, & \text{at irregular grid points,} \end{cases}$$

with prescribed Dirichlet boundary condition,

$$\Phi_{i,j} = w_{i,j} \qquad \text{on } \partial\mathcal{B},$$

to obtain approximate values of Φ at all grid nodes in $\mathcal{B}$. Here, the values of $w_{i,j}$ are obtained by evaluating the integral (3.5). The inversion of the discrete Poisson equation can be done with FFT's. According to [5] the error in the integrals at the irregular points is about $\mathcal{O}(\delta^3)$ if the smoothing parameter δ is not small relative to the grid size. This makes the local truncation error of the discrete Poisson equation at irregular grid nodes is of first-order, assuming $\delta = \mathcal{O}(h)$. The theory developed in [6] predicts that the resulting error in the values at the regular points should be about $\mathcal{O}(\delta^2)$.

Rather than use the extension method of step (3), we could compute the integrals (3.5) for each grid node directly, as in step (2), and we expect the result would be somewhat more accurate. In principle, the computational work is roughly of the order of the number of grid nodes with either procedure. However, we expect the approach in (3) takes less work since it is much more direct.

7 Numerical results

In this section, we present numerical results to illustrate the methods just described for the interface problem (2.1a)-(2.4). The interface Γ in the examples may have multiple disjoint components. Let K be the number of interface components. Each component Γ_k is assumed to be an ellipse Γ_k, for $k \in \{1,2,\cdots,K\}$. For ellipse Γ_k, its major and minor axes are denoted by a_k and b_k; the coordinates of its center are denoted by $(c_{k,1}, c_{k,2})$.

The known functions, V_m and J_m, and the vector $\mathbf{E}$ in the interface problem (2.1a)-(2.4) are chosen such that the exact solution of the interface problem is given by

$$\Phi_i(\mathbf{x}) = -\frac{\sigma_e}{\sigma_i + \sigma_e}\, x_1, \qquad \mathbf{x} = (x_1, x_2) \in \Omega_i,$$

and

$$\Phi_e(\mathbf{x}) = -x_1 - \frac{\sigma_e}{\sigma_i + \sigma_e} \sum_{k=1}^{K} \frac{a_k^2 (x_1 - c_{k,1})}{(x_1 - c_{k,1})^2 + (x_2 - c_{k,2})^2}, \qquad \mathbf{x} = (x_1, x_2) \in \Omega_e, \tag{7.1}$$

which indicates the field vector $\mathbf{E}$ is given by $\mathbf{E} = (1,0)^{\mathrm{T}}$. The conductivities are fixed to be $\sigma_e = 2$ and $\sigma_i = 1$. The rectangular box which embeds the interface in each example is the square $-1.5 \le x_1, x_2 \le 1.5$.

The boundary integral equation (3.7) for the interface problem is solved with the simple iteration (3.8). For all examples, the parameter β in the simple iteration is set to be $\beta = 1/(1+\mu)$. The unknown densities for the iteration are all initialized with zeros. The iteration stops when the difference between two successively iterated densities $\mathbf{q}_n$ in the scaled discrete ℓ^2-norm is less than tolerance $\epsilon_{\text{itr}} = 10^{-12}$, i.e., $\|\mathbf{q}_{n+1} - \mathbf{q}_n\|_2 < \epsilon_{\text{itr}}$. Here, the

scaled discrete l^2-norm of a vector $\mathbf{v}=(v_1,v_2,\cdots,v_J)^T\in\mathbb{R}^J$ is defined by

$$\|\mathbf{v}\|_2\equiv\sqrt{\frac{1}{J}\sum_{j=1}^{J}v_j^2}.$$

Since the exact solutions to the test problems are all known, the errors are directly computed by comparing the numerical and exact ones. The maximum errors of the numerical solutions at the irregular grid nodes and regular grid nodes are reported in Tables 1, 3 and 5 for each test problem with various grid sizes and numerical parameters. In the tables, C denotes the capacity parameter of the quad-tree node in the fast multipole summation; N denotes the number of grid elements along one direction; M_k denotes the number of points on each component of the interface (when the interface has multiple disconnected components, each component is represented by the same number of points); $\|e^{\text{irreg}}\|_\infty$ represents the maximum of the errors of the computed values of the boundary integral(s) at the irregular grid nodes; $\|e^{\text{reg}}\|_\infty$ is the maximum error at the regular grid nodes of the numerical solution to the Poisson equation.

In the experiments, we determine the smoothing parameter δ by choosing constant γ, which is also called a smoothing parameter, and setting

$$\delta=\gamma\frac{2\pi r_{\min}}{M_k}.$$

so that δ is on the order of $h_k=\mathcal{L}_k/M_k$ for each interface component. Here, $r_{\min}$ is the minimum of the semi-axes of the ellipse or all ellipses if the interface consists of multiple ellipses.

The computer codes for the numerical examples were written in the C++ computer language and are available to the readers upon request. The programs were run in a desktop computer which has a 2.8GHz Intel Xeon processor.

Example 7.1. In this example, the interface is one ellipse. Fig. 3 shows isolines of a numerical solution to the interface problem. Table 1 contains errors of the numerical solutions with different capacity and smoothing parameters. The results show that the expected third order accuracy at irregular grid nodes and second order accuracy at regular grid nodes are observed only when the capacity and smoothing parameters are sufficiently large. In particular, in the case that $\gamma=3$ and $C=10$, no convergence is evident, as the capacity parameter C is so small that the approximation of the regularized kernel by the standard one in the fast multipole summation introduces too much error. Table 1 also lists the CPU times used by the computer program for runs with different capacity and smoothing parameters. The timing results in Table 1 show that the computational work by this method on the grids with moderate sizes is roughly proportional to the number of unknowns or grid nodes around the cell boundary. This is because that the computer time used for the Poisson equation with FFTs is negligible as compared to that for the solution of the boundary integral equation and the evaluation of boundary integrals at

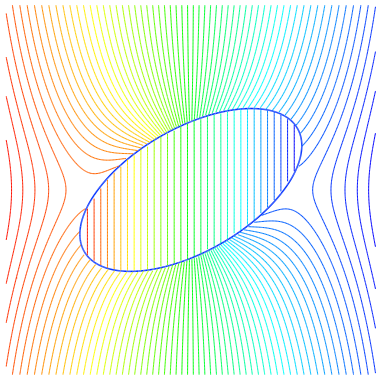


Figure 3: Isolines of a numerical solution to the interface problem with one ellipse.

Table 1: Numerical errors for the interface problem with one cell (Example 7.1).

C	γ	$N=M_k$	64	128	256	512	1024	2048
10	3	$\|e_h^{\text{irreg}}\|_\infty$	1.73E-3	3.71E-3	4.84E-3	4.61E-5	5.18E-3	6.81E-3
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	4.66E-1	7.67E-1	1.05E+2	8.90E-3	7.61E-1
		$\|e_h^{\text{reg}}\|_\infty$	2.41E-3	2.61E-3	6.41E-3	6.02E-5	9.60E-3	1.02E-2
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	9.23E-1	4.07E-1	1.06E+2	6.27E-3	9.41E-1
		t_{cpu} (secs)	0.050	0.12	0.28	0.62	1.4	3.5
20	3	$\|e_h^{\text{irreg}}\|_\infty$	1.73E-3	2.34E-4	2.70E-5	3.23E-6	4.15E-7	1.58E-6
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	7.39	8.67	8.36	7.78	2.63E-1
		$\|e_h^{\text{reg}}\|_\infty$	2.41E-3	6.69E-4	1.99E-4	5.10E-5	1.32E-5	3.37E-6
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	3.60	3.36	3.90	3.86	3.92
		t_{cpu} (secs)	0.050	0.13	0.26	0.55	1.4	3.3
40	3	$\|e_h^{\text{irreg}}\|_\infty$	1.73E-3	2.34E-4	2.70E-5	3.22E-6	4.15E-7	6.55E-8
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	7.39	8.67	8.39	7.76	6.34
		$\|e_h^{\text{reg}}\|_\infty$	2.41E-3	6.69E-4	1.99E-4	5.08E-5	1.32E-5	3.36E-6
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	3.60	3.36	3.92	3.85	3.93
		t_{cpu} (secs)	0.060	0.17	0.41	0.66	1.5	3.7
40	5	$\|e_h^{\text{irreg}}\|_\infty$	6.47E-3	9.16E-4	1.20E-4	1.50E-5	1.87E-6	2.46E-7
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	7.06	7.63	8.00	8.02	7.60
		$\|e_h^{\text{reg}}\|_\infty$	4.85E-3	1.44E-3	4.28E-4	1.13E-4	3.00E-5	7.72E-6
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	3.37	3.36	3.79	3.77	3.89
		t_{cpu} (secs)	0.060	0.17	0.40	0.66	1.5	3.7

irregular grid nodes. Table 2 lists the CPU times in seconds that are used for the discrete Poisson equation with FFTs on different grids. The first row in the table shows the number of grid elements along one direction.

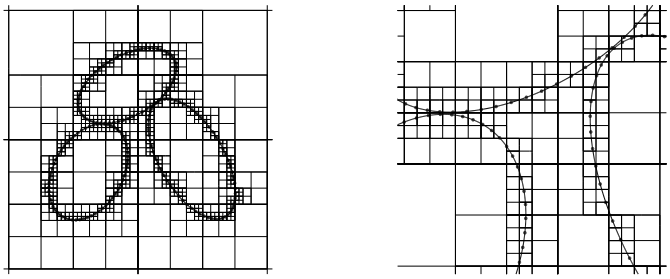
Table 2: CPU Times used for the Poisson equation with FFTs.

N	64	128	256	512	1024	2048	4096
t_{cpu} (secs)	0	0	0.01	0.04	0.18	0.85	3.64

Example 7.2. In this example the interface consists of three ellipses. The first ellipse is centered at point $(c_{1,1},c_{1,2})^{\mathrm{T}}=(0.625,-0.225)^{\mathrm{T}}$ with semi-axes $(a_1,b_1)^{\mathrm{T}}=(0.775,0.375)^{\mathrm{T}}$ and rotation angle $\theta=-60^\circ$. The second ellipse is centered at $(c_{2,1},c_{2,2})^{\mathrm{T}}=(-0.125,0.625)^{\mathrm{T}}$ with semi-axes $(a_2,b_2)^{\mathrm{T}}=(0.625,0.35)^{\mathrm{T}}$ and rotation angle $\theta=30^\circ$. The third ellipse is centered at $(c_{3,1},c_{3,2})^{\mathrm{T}}=(-0.575,-0.375)^{\mathrm{T}}$ with semi-axes $(a_3,b_3)^{\mathrm{T}}=(0.6,0.4)^{\mathrm{T}}$ and rotation angle $\theta=60^\circ$. These three ellipses are chosen to be very close to each other but not intersecting. The closest distance between the ellipses is about 0.006. See Fig. 4 for the ellipses and a close-up of the interface. Fig. 5 shows isolines of a numerical solution to the interface problem. Table 3 contains errors of the numerical solutions with different parameters. The results in Table 3 are consistent with those in the previous example. The expected third order accuracy at irregular grid nodes and second order accuracy at regular grid nodes are observed only when the capacity and smoothing parameters are sufficiently

Table 3: Numerical errors for the interface problem with three cells (Example 7.2).

C	γ	$N=M_k$	64	128	256	512	1024	2048
10	3	$\|e_h^{\mathrm{irreg}}\|_\infty$	5.80E-3	1.16E-3	1.65E-3	1.69E-3	1.74E-3	6.93E-3
		$\|e_{2h}^{\mathrm{irreg}}\|_\infty/\|e_h^{\mathrm{irreg}}\|_\infty$	X	5.00	7.03E-1	9.76E-1	9.71E-1	2.51E-1
		$\|e_h^{\mathrm{reg}}\|_\infty$	6.31E-3	1.44E-3	1.32E-3	2.63E-3	4.58E-3	1.20E-2
		$\|e_{2h}^{\mathrm{reg}}\|_\infty/\|e_h^{\mathrm{reg}}\|_\infty$	X	4.38	1.09	5.02E-1	5.74E-1	3.82E-1
		t_{cpu} (secs)	0.36	0.68	1.3	2.5	5.3	12.0
20	3	$\|e_h^{\mathrm{irreg}}\|_\infty$	5.80E-3	8.16E-4	1.02E-4	1.41E-5	2.17E-6	2.94E-7
		$\|e_{2h}^{\mathrm{irreg}}\|_\infty/\|e_h^{\mathrm{irreg}}\|_\infty$	X	7.11	8.00	7.23	6.50	7.38
		$\|e_h^{\mathrm{reg}}\|_\infty$	6.01E-3	1.28E-3	3.38E-4	1.02E-4	2.86E-5	6.80E-6
		$\|e_{2h}^{\mathrm{reg}}\|_\infty/\|e_h^{\mathrm{reg}}\|_\infty$	X	4.70	3.79	3.31	3.57	4.21
		t_{cpu} (secs)	0.39	0.71	1.2	2.3	4.8	11.0
40	3	$\|e_h^{\mathrm{irreg}}\|_\infty$	5.80E-3	8.16E-4	1.02E-4	1.41E-5	2.17E-6	2.94E-7
		$\|e_{2h}^{\mathrm{irreg}}\|_\infty/\|e_h^{\mathrm{irreg}}\|_\infty$	X	7.11	8.00	7.23	6.50	7.38
		$\|e_h^{\mathrm{reg}}\|_\infty$	6.01E-3	1.28E-3	3.38E-4	1.02E-4	2.86E-5	6.78E-6
		$\|e_{2h}^{\mathrm{reg}}\|_\infty/\|e_h^{\mathrm{reg}}\|_\infty$	X	4.70	3.79	3.31	3.57	4.22
		t_{cpu} (secs)	0.43	1.0	1.8	2.7	5.8	12.0
40	5	$\|e_h^{\mathrm{irreg}}\|_\infty$	2.32E-2	3.73E-3	4.85E-4	5.89E-5	8.63E-6	1.30E-6
		$\|e_{2h}^{\mathrm{irreg}}\|_\infty/\|e_h^{\mathrm{irreg}}\|_\infty$	X	6.22	7.69	8.23	6.83	6.64
		$\|e_h^{\mathrm{reg}}\|_\infty$	2.19E-2	3.57E-3	6.52E-4	1.85E-4	5.37E-5	1.48E-5
		$\|e_{2h}^{\mathrm{reg}}\|_\infty/\|e_h^{\mathrm{reg}}\|_\infty$	X	6.13	5.48	3.52	3.45	3.63
		t_{cpu} (secs)	0.45	1.0	1.8	2.6	5.7	12.0



(a) (b)

Figure 4: Adaptive grids for the fast multipole summation (three cells).

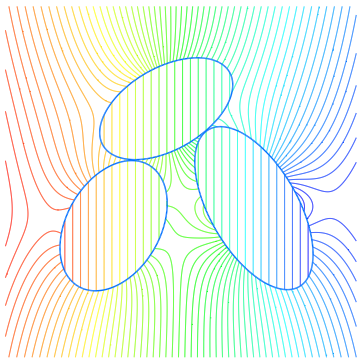
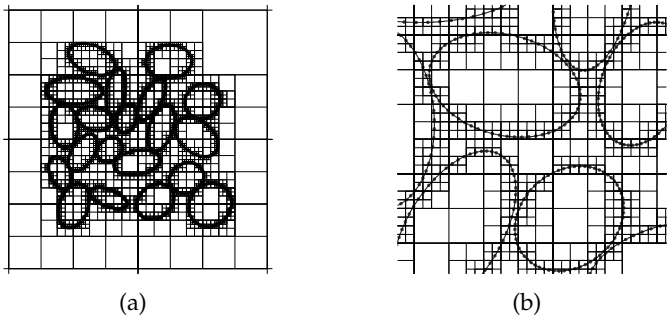


Figure 5: Isolines of a numerical solution to the interface problem with three ellipses.

large, but not with $\gamma=3$ and $C=10$. The run times are again proportional to the number of irregular grid nodes, instead of all grid nodes, since the CPU time used for the Poisson equation is negligible compared to that for the solution of the boundary integral equation and the evaluation of boundary integrals.

Example 7.3. In this example, the interface consists of twenty ellipses, whose axes and centers are listed in Table 4. The ellipses and a close-up are illustrated in Fig. 6. Fig. 7



(a) (b)

Figure 6: Adaptive grids for the fast multipole summation (twenty cells).

Table 4: Twenty ellipses for Example 7.3: major axis a_k, minor axis b_k, center $(c_{k,1},c_{k,2})$ and rotation angle θ (in degree).

k	a_k	b_k	$c_{k,1}$	$c_{k,2}$	θ
1	0.216669	0.141285	−0.526154	0.25008	165.569
2	0.244085	0.193709	0.210888	−0.749594	37.818
3	0.28455	0.115038	−0.269197	0.572747	81.4637
4	0.29436	0.192655	0.708311	0.030127	141.49
5	0.284289	0.144615	−0.0103632	−0.278066	9.396
6	0.250819	0.133865	−0.668486	−0.161434	58.735
7	0.273781	0.246768	0.871021	−0.79315	47.446
8	0.218416	0.205301	0.780105	0.461611	47.5849
9	0.341714	0.162392	−0.773844	0.588366	179.192
10	0.276414	0.174403	−0.77664	−0.802865	70.189
11	0.237011	0.101913	−0.358575	−0.709595	159.004
12	0.180211	0.117365	−0.135826	0.253291	68.7521
13	0.278021	0.202428	0.372457	0.937647	4.0367
14	0.263711	0.163481	−0.889039	0.162802	95.2703
15	0.165073	0.13415	−0.34441	−0.126333	39.7459
16	0.212647	0.177712	0.588054	−0.466227	25.219
17	0.27797	0.117366	0.167368	0.484899	60.798
18	0.204751	0.124421	−0.94228	−0.403608	110.762
19	0.289704	0.142591	0.280349	0.135601	71.381
20	0.294926	0.168056	−0.558163	0.968711	157.659

shows isolines of a numerical solution to the interface problem. Table 5 contains errors of the numerical solutions with different parameters. The results are consistent with the previous two examples. With large and compatible capacity and smoothing parameters

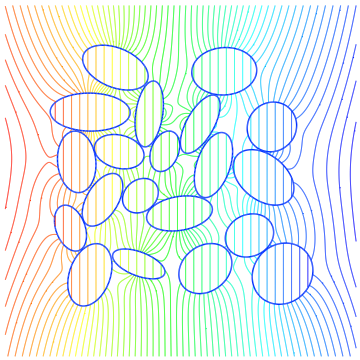


Figure 7: Isolines of a numerical solution to the interface problem with twenty ellipses.

Table 5: Numerical errors for the interface problem with twenty cells (Example 7.3).

C	γ	$N=M_k$	64	128	256	512	1024	2048
10	3	$\|e_h^{\text{irreg}}\|_\infty$	9.36E-3	1.20E-2	9.52E-3	9.88E-3	1.06E-2	1.07E-2
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	7.80E-1	1.26	9.64E-1	9.32E-1	9.91E-1
		$\|e_h^{\text{reg}}\|_\infty$	9.84E-3	2.37E-2	4.80E-2	1.10E-1	1.12E-1	2.65E-1
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	4.15E-1	4.94E-1	4.36E-1	9.82E-1	4.23E-1
		t_{cpu} (secs)	3.3	6.0	11.	22.	44.	93.
20	3	$\|e_h^{\text{irreg}}\|_\infty$	1.25E-3	1.88E-4	2.89E-5	3.63E-6	2.60E-6	2.78E-6
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	6.65	6.51	7.96	1.40	9.35E-1
		$\|e_h^{\text{reg}}\|_\infty$	6.76E-3	3.11E-3	7.08E-4	1.74E-4	4.27E-5	1.84E-5
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	2.17	4.39	4.07	4.07	2.32
		t_{cpu} (secs)	3.3	5.2	9.2	18.	35.	74.
40	3	$\|e_h^{\text{irreg}}\|_\infty$	1.25E-3	1.88E-4	2.89E-5	3.48E-6	4.08E-7	8.51E-8
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	6.65	6.51	8.30	8.53	4.79
		$\|e_h^{\text{reg}}\|_\infty$	6.76E-3	3.12E-3	7.09E-4	1.74E-4	4.44E-5	1.11E-5
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	2.17	4.40	4.07	3.92	4.00
		t_{cpu} (secs)	4.8	7.8	11.	19.	37.	76.
40	5	$\|e_h^{\text{irreg}}\|_\infty$	5.89E-3	8.70E-4	1.30E-4	1.68E-5	1.95E-6	2.69E-7
		$\|e_{2h}^{\text{irreg}}\|_\infty/\|e_h^{\text{irreg}}\|_\infty$	X	6.77	6.69	7.74	8.62	7.25
		$\|e_h^{\text{reg}}\|_\infty$	1.33E-2	3.03E-3	6.80E-4	1.70E-4	4.45E-5	1.10E-5
		$\|e_{2h}^{\text{reg}}\|_\infty/\|e_h^{\text{reg}}\|_\infty$	X	4.39	4.46	4.00	3.82	4.05
		t_{cpu} (secs)	4.7	7.6	11.	19.	36.	75.

($C=40$, $\gamma=3$ or $\gamma=5$), third order accuracy at irregular grid nodes and second order accuracy at regular grid nodes are observed. The timing results again show linear growth.

8 Discussion

This work describes a boundary integral method for potentials on closely packed cells. When portions of the boundary are close to each other, the boundary integrals become nearly singular. The nearly singular integrals are evaluated by a regularization of the integral kernel which admits analytically determined corrections to maintain accuracy. To speed up the dense matrix vector product associated with the boundary integrals, the fast multipole method is used. The combination of the fast multipole method and the regularized boundary integral makes the method fast as well as accurate. However, since the fast multipole method only works with the standard integral kernel instead of the regularized one, the approximation of the regularized kernel by the standard one in the far field computation of the fast multipole method introduces extra errors. To minimize the approximation errors, the capacity and smoothing parameters need to be appropriately selected. Numerical examples for the interface problem show that, when the capac-

ity and smoothing parameters are sufficiently large, the boundary integral method presented here yields high order accurate solutions, and the computational work is roughly proportional to the number of irregular grid nodes around the interface on grids with moderately large sizes.

In the numerical experiments, we have not fine-tuned the parameter β for the best performance of the simple iteration. With the chosen value for the parameter, a typical number of iterations for the boundary integral equation is around 30, corresponding to the iteration tolerance $\epsilon_{\text{itr}} = 10^{-12}$. If a Krylov subspace method such as the GMRES method is used for the boundary integral equation, we expect that the iteration number will be reduced roughly by a half [17]. In this way, the method will become much more efficient.

Our numerical experiments indicate that the iteration number is not sensitive to the closeness of the cells. However, at the moment, it is probably difficult to say how the rate of convergence of the iteration depends on the geometry of the interfaces.

It is possible that the fast multipole method could be replaced with other fast summation techniques such as the Barnes-Hut algorithm [3] so that both near field and far field computations could be done with the regularized kernel. In this way, the fine-tuning of the capacity and smoothing parameters might be avoided. The extension of the method to three space dimensions is straightforward. The corresponding work will be reported separately.

Acknowledgments

The authors are grateful to the referees for their constructive comments which led to improvement in presentation of this paper. Research of the first author was supported in part by the National Science Foundation of the USA under Grant DMS–0915023, the National Natural Science Foundation of China under Grant DMS–11101278 and the Young Thousand Talents Program of China. Research of the second author was supported in part by the National Science Foundation of the USA under Grant DMS–0806482.

Appendix

In deriving the integral equation (3.6) we will need the normal derivative of the double layer potential such as (3.1) on Γ. For u as in (3.1), $\partial u/\partial \mathbf{n}$ is continuous at Γ, and

$$\frac{\partial u(\mathbf{x})}{\partial \mathbf{n}_{\mathbf{x}}} = \int_{\Gamma} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{t}_{\mathbf{x}}} \frac{\partial f(\mathbf{y})}{\partial \mathbf{t}_{\mathbf{y}}} ds_{\mathbf{y}}. \tag{A.1}$$

This is a classical fact (e.g. see [11, p. 5] for this formula, or [8, Thm. 2.23, p. 57] for the 3D case), but we sketch the derivation for completeness. Starting with the integral for $u(\mathbf{x})$ in (3.1), for $\mathbf{x} \notin \Gamma$, we can apply $\nabla_{\mathbf{x}} = -\nabla_{\mathbf{y}}$ to G inside the integral, use $\Delta_{\mathbf{y}} G = 0$ to convert

the normal derivative to a tangential derivative, and integrate by parts to obtain

$$\nabla u(\mathbf{x}) = \int_\Gamma \left(\frac{\partial}{\partial x_2}, -\frac{\partial}{\partial x_1} \right) G(\mathbf{x}-\mathbf{y}) f'(\mathbf{y}) ds_\mathbf{y}, \qquad \mathbf{x} \notin \Gamma,$$

where $f'(\mathbf{y}(s)) = (\partial/\partial s) f(\mathbf{y}(s))$. Now suppose we extend $\mathbf{t}$ and $\mathbf{n}$ from Γ to a neighborhood as orthogonal vector fields. Then

$$\mathbf{n}_\mathbf{x} \cdot \nabla \mathbf{u}(\mathbf{x}) = \int_\Gamma \mathbf{t}_\mathbf{x} \cdot \nabla_\mathbf{x} G(\mathbf{x}-\mathbf{y}) f'(\mathbf{y}) ds_\mathbf{y}, \qquad \mathbf{x} \notin \Gamma.$$

Since the tangential gradient of the single layer potential is continuous at Γ, we can now let $\mathbf{x}$ approach Γ and obtain (A.1).

Now to derive (3.6), we find the normal derivatives of Φ_i and Φ_e at Γ, applying (3.3) and (A.1) to (3.5). We find that

$$\frac{\partial \Phi_i(\mathbf{x})}{\partial \mathbf{n}_\mathbf{x}} = \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{t}_\mathbf{x}} \frac{\partial V_m(\mathbf{y})}{\partial \mathbf{t}_\mathbf{y}} ds_\mathbf{y} - \int_\Gamma \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial \mathbf{n}_\mathbf{x}} q(\mathbf{y}) ds_\mathbf{y} + \frac{1}{2} q(\mathbf{x}) + \mathbf{E} \cdot \mathbf{n}_\mathbf{x}$$

and $\partial \Phi_e(\mathbf{x})/\partial \mathbf{n}_\mathbf{x}$ is the same except that the term $\frac{1}{2}q$ is replaced by $-\frac{1}{2}q$. Substituting these two expressions into (2.3) and rearranging, we obtain (3.6).

References

[1] T. Ashihara, T. Yao, T. Namba, M. Ito, T. Ikeda, A. Kawase, S. Toda, T. Suzuki, M. Inagaki, M. Sugimachi, M. Kinoshita, K. Nakazawa, Electroporation in a model of cardiac defibrillation, J. Cardiovasc. Electrophysiol. 12 (12) (2001) 1393–1403.

[2] K. E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, Cambridge, UK, 1997.

[3] J. Barnes, P. Hut, A hierarchical $O(N \log N)$ force calculation algorithm, Nature 324 (2) (1986) 446–449.

[4] J. T. Beale, A grid-based boundary integral method for elliptic problems in three-dimensions, SIAM J. Numer. Anal. 42 (2004) 599–620.

[5] J. T. Beale, M. C. Lai, A method for computing nearly singular integrals, SIAM J. Numer. Anal. 38 (6) (2001) 1902–1925.

[6] J. T. Beale, A. T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, Commun. Appl. Math. Comput. Sci. 1 (2006) 91–119.

[7] B. A. Cipra, The best of the 20th century: Editors name top 10 algorithms, SIAM News 33 (4) (2000) 2.

[8] D. Colton, R. Kress, Integral Equation Methods in Scattering Theory, Wiley, New York, 1983.

[9] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.

[10] J. T. Hamilton, G. Majda, On the rokhlin-greengard method with vortex blobs for problems posed in all space or periodic in one direction, J. Comput. Phys. 121 (1995) 29–50.

[11] G. C. Hsiao, W. L. Wendland, Boundary Integral Equations, Springer, 2008.

[12] W. Krassowska, J. C. Neu, Response of a single cell to an external electric field, Biophysical Journal 66 (1994) 1768–1776.

[13] A. Mayo, Fast high order accurate solution of Laplace's equation on irregular regions, SIAM J. Sci. Statist. Comput. 6 (1985) 144–157.
[14] L. M. Mir, M. F. Bureau, J. Gehl, R. Rangara, D. Rouy, J. M. Caillaud, P. Delaere, D. Branellec, B. Schwartz, D. Scherman, High-efficiency gene transfer into skeletal muscle mediated by electric pulses, Proc. Natl. Acad. Sci. USA 96 (8) (1999) 4262–4267.
[15] E. Neumann, S. Kakorin, K. Toensing, Fundamentals of electroporative delivery of drugs and genes, Bioelectrochem. Bioenerg. 48 (1) (1999) 3–16.
[16] C. Pozrikidis, Interfacial dynamics for stokes flow, J. Comput. Phys. 169 (2001) 250–301.
[17] Y. Saad, Iterative methods for sparse linear systems, PWS Publishing Company, Boston, 1996.
[18] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual method for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[19] S. A. Sauter, C. Schwab, Boundary Element Methods, Springer Series in Computational Mathematics 39, 2011.
[20] G. Sersa, T. Cufer, M. Cemazar, M. Rebersek, R. Zvonimir, Electro-chemotherapy with bleomycin in the treatment of hypernephroma metastasis: case report and literature review, Tumori 86 (2) (2000) 163–165.
[21] H. A. van der Vorst, BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution nonsymmetric linear systems, SIAM J. Sci. Stat. 13 (1992) 631–644.
[22] S. K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2d, J. Comput. Phys. 228 (2009) 2334–2353.
[23] S. K. Veerapaneni, A. Rahimian, G. Biros, D. Zorin, A fast algorithm for simulating vesicle flows in three dimensions, J. Comput. Phys. 230 (2011) 5610–5634.
[24] L. Ying, G. Biros, D. Zorin, A high-order 3d boundary integral equation solver for elliptic PDEs in smooth domains, Journal of Computational Physics 219 (1) (2006) 247–275.
[25] W.-J. Ying, C. S. Henriquez, Hybrid finite element method for describing the electrical response of biological cells to applied fields, IEEE Trans. Biomed. Engrg. 54 (4) (2007) 611–620.
[26] A. Z. Zinchenko, R. H. Davis, An efficient algorithm for hydrodynamical interaction of many deformable drops, J. Comput. Phys. 157 (2000) 539–587.
[27] A. Z. Zinchenko, M. A. Rother, R. H. Davis, A novel boundary-integral algorithm for viscous interaction of deformable drops, Phys. Fluids 9 (1997) 1493–1511.